

Laboratorio Equipos y Sistemas de Control Digital
Guía Laboratorio Nº 1

“Programación en C para plataforma CADmega128, basada en microcontrolador Atmel megaAVR”

En esta experiencia se verá en forma aplicada la manipulación e implementación de las interfaces digitales del módulo CADmega128 a través de programación en C.

1. Objetivos

- Introducción básica de programación en C para plataformas basadas en microcontroladores.
- Manejo de las entradas digitales del módulo CADmega128 asociadas a los pulsadores (SW1 a SW5) mediante programación en C.
- Manipular las salidas digitales, del módulo CADmega128 asociadas a los leds (LED_G0 a LED_G2) y buzzer (BZ1) a través de rutinas de programación en C.

2. Exigencias del Preinforme

Responder brevemente las siguientes preguntas relacionadas con conceptos básicos de programación en C para el módulo CADmega128.

2.1 Especifique las características técnicas y periféricos (botones, pantalla, indicadores LED, interfaces de comunicación serie, puertos de expansión, interfaces ADC y DAC) del módulo CADmega128.

2.2 ¿Qué es la interfaz JTAG y como se utiliza en programación de microcontroladores Atmel AVR?.

3. Desarrollo

Materiales y equipamientos necesarios para la experiencia.

- 1 módulo CADmega128 con fuente de poder
- 1 programador AVR JTAG ICE USB con cable de interfaz JTAG.
- PC IBM compatible o notebook con S.O. Windows XP/7.

Actividades previas sobre el Hardware.

1. Siga atentamente las instrucciones e indicaciones que el profesor entregará sobre el uso y manipulación del módulo CADmega128. Consulte atentamente también el **Manual de Usuario** del módulo, en caso de cualquier duda que tenga sobre el uso y conexión de entradas y salidas del equipo.

2. Se requiere que el computador tenga previamente instalado el entorno de programación AVR Studio 4.19 de Atmel y el *toolchain* WinAVR.

3. Además se debe instalar el driver del programador AVR JTAG ICE USB en el computador. Luego verificar el número del puerto COM asociado al puerto virtual instalado por el driver.

4. Encender computador. Conectar al computador el programador AVR JTAG ICE a un puerto USB y luego conectar el cable ribbon al conector JTAG de la placa CADmega 128.

Actividades previas sobre el Software.

1. Abrir el software AVR Studio 4.19 y crear un proyecto nuevo. Asociar el proyecto al microcontrolador ATMEGA128 y el grabador AVR JTAG ICE.

2. Cargar el código base (CADmega128-base.c). Este archivo junto con las librerías asociadas serán facilitadas por el profesor y contiene todas las llamadas a librerías externas y definición de macros necesarias para el proyecto.

3. Compruebe la conexión y la comunicación del grabador AVR JTAG ICE en el software AVR Studio.

Actividades de la experiencia.

Desarrolle un programa que permita las siguientes interacciones con los LEDS:

a) Destello de LED_G0 con intervalo de un segundo.
b) Mostrar la cuenta ascendente en binario a tres bits (escriba directo una variable tipo *char* al puerto G).
c) Mostrar el encendido y apagado secuencial de LED_G0, LED_G1 y LED_G2 con intervalo de 500ms.

El código para trabajar con los LEDs tiene el siguiente formato:

```
set_led(0,1); // Enciende el LED G0 (izquierda)
set_led(2,0); // Apaga el LED G2 (derecha)
```

Desarrolle un programa que permita las siguientes interacciones con los botones y los LEDS:

a) Al presionar SW1 se enciende LED_G0, al presionar SW2 se apaga LED_G0. Al presionar SW3 se enciende LED_G1, al presionar SW4 se apaga LED_G1.

b) Al presionar SW1 comienza el destello de todos los LEDS con intervalo de un segundo y se mantiene de ese modo en forma infinita.

c) Al presionar SW1 mostrar la cuenta ascendente en binario a tres bits. Luego al presionar SW2 se detiene la ejecución de la cuenta. Al presionar SW3 muestra la cuenta en forma descendente.

El código para la lectura de los botones tiene el siguiente formato:

```
if (sw_state(num_de_switch)==1) { código cuando el botón se presiona… }
else { opcional, código si el botón NO se presiona… }
```

Por ejemplo, para activar el LED G1 cuando se presione SW3 (conectado al bit 4 del puerto D), se puede implementar a través de:

```
if (sw_state(3)==1) set_led(1,1); // si SW3 está presionado, encender LED
G1.
```

Si se desea leer los switches mediante flancos, útil para evitar repeticiones indeseadas de los botones, puede usarse la función:

`sw_flanco(num_de_switch)`, incluida en el Proyecto Base. La función retorna valor tipo unsigned char: 1 si el botón acaba de ser presionado

(flanco positivo), 0 si mantiene el mismo estado anterior; la función sólo revisa al momento de ser llamada.

Por ejemplo, para incrementar una variable k (previamente declarada) cada vez que se presione SW2, basta con el código:

```
loop:
if (sw_flanco(2)) { k++; }
goto loop;
```

Desarrolle un programa que permita las siguientes interacciones con los botones, los LEDS y el buzzer:

a) Generar un tono de frecuencia constante en el buzzer al iniciarse el programa y mientras de presiona SW1. Probar con diferentes tonos y diferentes duraciones de tiempos para los sonidos.

b) Al presionar SW1 se enciende LED_G0 y generar un tono agudo en el buzzer, al presionar SW2 generar un tono grave en el buzzer se apagar LED_G0.

c) Al presionar SW1 genera una secuencia de tonos ascendentes en el buzzer durante aproximadamente un segundo. Luego al presionar SW2 genera una secuencia de tonos descendentes en el buzzer durante aproximadamente un segundo.

El código base para trabajar con el buzzer tiene el siguiente formato:

```
void buzzer()
{ char i;
for (i=0; i<100; i++)
{ _delay_us(200); // pausa de 200 uS
sbi(PORTF,3); // 1->PF3
_delay_us(200); // pausa de 200 uS
cbi(PORTF,3); // 0->PF3
}
       }
```

La configuración de los terminales de los puertos G y F para trabajar con los LEDs y el buzzer está incluido el código base.

4. Requerimientos del Informe

El informe se deberá entregar dentro de dos semanas de realizada la experiencia, impreso y en formato digital, junto con los archivos fuentes generados para la experiencia. El informe contendrá:

4.1. Introducción (Características de C para microcontroladores, no más de 2 Hojas).
4.2. Descripción de cada una de las rutinas implementadas dentro de la experiencia.
4.3. Programas generados en la experiencia, con comentarios en código fuente de cada programa (sólo formato digital).

4.4. Conclusiones individuales.

5. Referencias

Información y manual de CADmega128 versión 1.2 en sitio www.automaticausach.cl

Santiago, Mayo de 2015